CLAIMS

WHAT IS CLAIMED IS:

1. A method of creating a computer program that uses a cryptographic algorithm to apply a cryptographic key to first data, said method comprising the acts of:

identifying a set of actions that are performed in the course of using said cryptographic algorithm to apply said cryptographic key to said first data;

generating a first set of computer-executable instructions which includes instructions to perform said actions;

including said first set of computer-executable instructions in said computer program, wherein said computer program does not require access to said cryptographic key.

2. The method of claim 1, wherein said cryptographic algorithm is a public/private-key algorithm.

3. The method of claim 2, wherein said cryptographic key is the private key of an asymmetric key pair.

4. The method of claim 1, further comprising the act of receiving second data which in some way identifies or relates to a computing device on which said computer program runs, and wherein said first set of computer-executable instructions is based on said second data.

5. The method of claim 4, wherein said second data comprises or is based on one or more of the following: a CPUID associated with a processor of said computing device; a serial number associated with said processor; and third data which identifies a hard disk associated with said computing device, said third data being assigned to said hard disk by a manufacturer or distributor of said hard disk.

6. The method of claim 4, wherein said first set of computer-executable instructions comprises one or more instructions which depend for their correct execution on the retrieval during execution of said second data.

7. The method of claim 1, further comprising the act of randomly or pseudo-randomly generating a number, wherein said first set of computer-executable instructions is based on said number.

8. The method of claim 1, further comprising the acts of:

generating a diversionary second set of computer-executable instructions which perform one or more second actions; and

including said second set of computer-executable instructions in said computer program.

9. The method of claim 8, further comprising the act of retrieving said diversionary second set of computer-executable instructions from a database of stored code.

10. The method of claim 8, wherein said computer program does not rely on performance of said second actions to apply said cryptographic key to said first data.

11. The method of claim 1, further comprising the act of generating a second set of computer-executable instructions which detects modification or deletion of at least a portion of code contained in said computer program, and which restores said portion if said portion has been deleted or modified.

12. The method of claim 1, further comprising the act of reorganizing at least some code contained in said computer program.

13. The method of claim 1, further comprising the acts of:

delimiting a segment of at least some code contained in said computer program;

obtaining a first hash of the code inside the delimited segment;

including said first hash of the delimited segment within said computer program;

creating a second set of computer-executable instructions which obtains a second hash of the delimited segment and which compares said second hash with said first hash; and

including said second set of computer-executable instructions in said computer program.

14. The method of claim 1, further comprising the acts of:

encrypting at least a portion of said first set of computer-executable instructions; and

creating a second set of computer-executable instructions which decrypts said portion.

15. The method of claim 1, wherein said act of creating said first set of computer-executable instructions comprises the acts of:

creating instructions in a source-level language; and

compiling said source-level-language instructions.

16. The method of claim 15, further comprising the act of postprocessing the compiled instructions after said compiling act, wherein said postprocessing act

comprises one or more of the following: encrypting at least a portion of the compiled instructions, and hashing at least a portion of the compiled instructions.

17. The method of claim 1, further comprising the acts of:

receiving, from a computing device, a request for said computer program via a network; and

providing said computer program to said computer device via said network.

18. The method of claim 17, wherein said network comprises the Internet.

19. The method of claim 17, wherein said receiving act occurs substantially contemporaneously with said providing act.

20. The method of claim 1, wherein said generating act comprises retrieving instructions from a database of stored code.

21. A computer-readable medium encoded with a third set of computer-executable instructions to perform the method of claim 1.

22. A method of securely decrypting data with a cryptographic key, said method comprising the acts of:

performing a first set of actions which apply said cryptographic key to said data, said first set of actions not requiring for their performance access to said cryptographic key; and

performing a diversionary second set of actions different from said first set of actions;

wherein said first and said second sets of actions are implemented by way of a set of computer-executable instructions executable on a computing device.

23. The method of claim 22, wherein performance of said first set of actions does not depend on performance of said diversionary second set of actions.

24. The method of claim 22, wherein either of said first or second sets of actions in some manner relies for its performance on retrieval or derivation from said computing device of hardware identification data which identifies or in some way relates to hardware associated with said computing device.

25. The method of claim 22, further comprises the acts of:

detecting a modification or deletion of at least a portion of said set of computer-executable instructions; and

restoring said set of instructions to its state prior to said modification or deletion.

26. The method of claim 22, further comprises the act of decrypting at least a portion of said set of computer-executable instructions prior to executing said portion.

27. The method of claim 26, further comprising the act of re-encrypting said portion subsequent to executing said portion.

28. The method of claim 22, further comprising the acts of:

deriving a value based on at least a portion of said set of computer-executable instructions; and

comparing the derived value to a stored value.

29. The method of claim 28, wherein said act of deriving comprises the act of hashing said portion.

30. The method of claim 22, further comprising the act of moving at least some of said computer-executable instructions to a randomly or pseudo-randomly selected memory location on said computing device prior to execution of the moved instructions.

31. A computer-readable medium encoded with said set of computer-executable instructions to perform the method of claim 22.

32. A method of performing an action on a computing device in a manner that is at least partly resistant to modification or analysis, said method comprising the acts of:

executing on said computing device a first set of computer-executable instructions that implements a sub-action, wherein performance of said action is in at least some way furthered by performance of said sub-action; and

executing on said computing device a second set of computer-executable instructions that implements said sub-action, said second set of computer-executable instructions being different from said first set of computer-executable instructions.

33. The method of claim 32, wherein said action comprises applying a cryptographic key to first data.

34. The method of claim 33, wherein said action comprises using said cryptographic key to decrypt said first data.

35. The method of claim 33, wherein said action comprises using said cryptographic key to authenticate said first data.

36. The method of claim 32, further comprising the act of executing a diversionary third set of computer-executable instructions different from said first and second sets of computer-executable instructions.

37. The method of claim 36, wherein neither said first or second sets of computer-executable instructions relies for its correct performance on said diversionary third set of computer-executable instructions.

38. The method of claim 32, further comprising the acts of:

detecting a modification or deletion of at least a portion of said first or second sets of computer-executable instructions; and

restoring the modified or deleted instructions to their state prior to said modification or deletion.

39. The method of claim 32, further comprises the act of decrypting at least a portion of said first or second sets of computer-executable instructions prior to executing said portion.

40. The method of claim 39, further comprising the act of encrypting said portion subsequent to executing said portion.

41. The method of claim 32, further comprising the acts of

deriving a value based on at least a portion of said first or second sets of computer-executable instructions; and

comparing the derived value to a stored value.

42. The method of claim 41, wherein said act of deriving comprises the act of hashing said portion.

43. The method of claim 32, further comprising the act of moving at least some of said first or second set of computer-executable instructions to a randomly or pseudo-randomly selected memory location prior to their execution on said computing device.

44. A computer-readable medium encoded with computer-executable instructions to perform the method of claim 32.

45. A method of creating a computer program that is at least partly resistant to modification or analysis wherein said computer program performs a first action on at least two different occasions, said method comprising the acts of:

creating a first set of computer-executable instructions which performs said first action;

including said first set of computer-executable instructions at a first location in said computer program;

creating a second set of computer-executable instructions which performs said first action, said second set of computer-executable instructions being at least in part different from said first set of computer-executable instructions; and

including said second set of computer-executable instructions at a second location in said computer program.

46. The method of claim 45, wherein said first location is inline with code that requires performance of said action.

47. The method of claim 45, wherein said first action comprises applying a cryptographic key to first data.

48. The method of claim 47, wherein performance of said first action does not require access to said cryptographic key.

49. The method of claim 45, further comprising the act of receiving second data which in some way identifies or relates to a computing device on which said computer program runs, and wherein said first set of computer-executable instructions is based on said second data.

50. The method of claim 45, further comprising the act of randomly or pseudo-randomly generating a number, wherein said first set of computer-executable instructions is based on said number.

51. The method of claim 45, further comprising the acts of:

creating a diversionary third set of computer-executable instructions; and

including said diversionary third set of computer-executable instructions in said computer program.

52. The method of claim 45, further comprising the act of creating a third set of computer-executable instructions which detects modification or deletion of at least a portion of said computer program, and which restores said portion to its state prior to modification or deletion.

53. The method of claim 45, further comprising the act of reorganizing said first or second sets computer-executable instructions or a combination thereof.

54. The method of claim 45, further comprising the acts of:

delimiting a segment of said computer program;

obtaining a first hash of the code inside the delimited segment;

including said first hash of the delimited segment within said computer program; and

creating a third set of computer-executable instructions which obtains a second hash of the delimited segment and which compares said second hash with said first hash.

55. The method of claim 45, further comprising the acts of:

encrypting at least some instructions in said computer program; and

creating a third set of computer-executable instructions which decrypts said encrypted instructions prior to their execution.

56. The method of claim 45, wherein said act of creating said first set of computer-executable instructions comprises:

creating instructions in a source-level language; and

compiling said source-level-language instructions.

57. The method of claim 56, further comprising the act of postprocessing the compiled instructions, wherein said postprocessing act comprises one or more of the following: encrypting at least a portion of the compiled instructions, and hashing at least a portion of the compiled instructions.

58. The method of claim 45, further comprising the acts of:

receiving, from a computing device, a request for said computer program via a network; and

providing said computer program to said computer device via said network;

59. The method of claim 58, wherein said network comprises the Internet.

60. The method of claim 58, wherein said receiving act occurs substantially contemporaneously with said providing act.

61. The method of claim 45, further comprising the act of retrieving instructions from a database of stored code to be included in said computer program.

62. A computer-readable medium encoded with a third set of computer-executable instructions to perform the method of claim 45.

63. A method of creating a computer program that is at least partly resistant to modification or analysis, said method comprising the acts of:

creating a first set of computer-executable instructions; and

creating a second set of computer-executable instructions which detects modification or deletion of at least a portion of said first set of computer-executable instructions and which restores said at least a portion if said at least a portion has been deleted or modified.

64. The method of claim 63, wherein said second set of computer-executable instructions perform a process comprising the acts of:

hashing at least a portion of the instructions in said computer program; and

comparing the result of said hashing act with a stored value.

65. The method of claim 63, further comprising the act of receiving first data which in some way identifies or relates to a computing device on which said computer program runs, and wherein said first or second set of computer-executable instructions is based on said first data.

66. The method of claim 63, further comprising the act of randomly or pseudo-randomly generating a number, wherein said first or second set of computer-executable instructions is based on said number.

67. The method of claim 63, further comprising the act of creating a diversionary third set of computer-executable instructions which perform one or more actions.

68. The method of claim 67, wherein said first and said second sets of computer-executable instructions do not rely for their correct execution on said diversionary third set of computer-executable instructions.

69. The method of claim 63, further comprising the acts of:

creating instructions in a source-level language; and

compiling the source-level-language instructions to produce said computer program.

70. The method of claim 63, further comprising the acts of:

encrypting at least some instructions in said computer program; and

creating a third set of computer-executable instructions which decrypts said encrypted instructions prior to their execution.

71. A computer readable medium comprising:

a first set of computer-executable instructions; and

a second set of computer-executable instructions which uses error-correction techniques to detect variations of said first set of computer-executable instructions from a reference state, and to restore said first set of computer-executable to said reference state.

72. The computer-readable medium of claim 71, wherein said reference state comprises the state of said first set of computer-executable instructions immediately after said computer-executable instructions are loaded into memory for execution.

73. The computer-readable medium of claim 71, wherein first set of computer-executable instructions are dynamically modifiable during their execution, and wherein said reference state comprises a state of said first set of computer-executable instructions at an intermediate point in time during their execution.